数理解析研究所講究録
第 848 巻 1993 年 100-108

Algorithm of solving second order linear ordinary differential equations and its implementation in REDUCE

L. M. Berkovich
Dept. Algebra and Geometry, Samara State University, Russia
F. L. Berkovich
Dept. Applied Mathematics, Samara State Avia-Space University, Russia

1. Introduction

The construction of algorithms for finding formal solutions for some classes of equations is a main purpose of any constructive theory of ordinary differential equations theory. Explicit formulas are the most important and conclude in itself all available information. It is necessary to have them also to convince in our physics intuition and to compare different theories including the bounds of their applicability.

Euler, Liouville, Kummer, Jacoby and other mathematicians discovered that the basic method of integration and investigation of differential equations is an introducing the convenient changes of variables which reduce the original equations to simpler form. However, they didn't propose the algorithms for finding this transformations and, as a result, search for suitable substitutions had an euristic character.

From the other hand, deep relations between linear ODE and algebraic equations have been known for a long time and that lead to the method of factorization of differential operators. Here many results had a non-constructive character, too.

One of the authors developed in [1] the effective method of transformation of differential equations in which there were used both change of variables and factorization of differential operators.

There exist the different approaches for receiving the Liouvillian solutions of ODE, i.e., those ones that may be represented in a finite form using elementary, algebraic functions and quadratures.

An interesting though not complete, review of different methods for receiving the Liouvillian and other formal solutions of ODE may be found in [2].

Modern computer algebra systems are becoming the powerful means of PC implementation of the exact methods of ODE investigating and integrating. As a result, many users are able now to use practically those methods available earlier for specialists only.

In this paper we propose our algorithm for finding the general solutions of nonhomogeneous second order LODE

$$Ly \equiv a_2(x)y'' + a_1(x)y' + a_0(x)y = f(x)$$

where coefficients a_2, a_1, a_0 belong to some differential field K and they are arbitrary differentiable functions, possibly, containing parameters. The heart of an algorithm is a search for variable change that reduces the correspondent homogeneous equation $Ly = 0$ to one with constant coefficients.

Algorithm is implemented in computer algebra system REDUCE.

2. Method

2.1. Search for the Kummer-Liouville transformation

Let the equation

$$y'' + a_1(x)y' + a_0(x)y = 0, a_1 \in \mathrm{C}^1(I), a_0 \in \mathrm{C}(I) \tag{1}$$

where $I = \{x|a < x < b\}$ be given. Here for simplicity we assumed $a_2 = 1$. Let us apply here the *Kummer-Liouville (KL) transformation* , i.e. the variable change:

$$y = v(x)z(t), \;\; dt = u(x)dx, \quad u, v \in \mathrm{C}^2(I), uv \neq 0, \forall x \in I. \tag{2}$$

Due to the Stäckel-Lie theorem (2) is a most general transformation that keeps the order and linearity of an equation (1). it is a basic part of the Kummer problem (see [3]) of reduction of (1) to the equation of a form

$$z''(t) + b_1(t)z'(t) + b_0(t)z(t) = 0, \quad b_1 \in \mathrm{C}^1(J), b_0 \in \mathrm{C}(J) \tag{3}$$

where $J = \{t|c < t < d\}$.

The Kummer problem is always solvable [4] and, therefore, there always exists the KL-transformation that reduces (1) to (3). However, the problem of reduction of (1) to the equation with constant coefficients

$$z''(t) + b_1 z(t) + b_0 z(t) = 0, \quad b_1, b_0 = const \tag{4}$$

is of the most essential interest.

The KL-transformation may be found using

Lemma 1. The equation (1) may be reduced to (4) by the KL-transformation, for which the kernel $u(x)$ satisfies to the second order Kummer-Schwarz equation

$$\frac{1}{2}\frac{u''}{u} - \frac{3}{4}(\frac{u'}{u})^2 - \frac{1}{4}\delta u^2 = A_0(x), \tag{5}$$

where

$$A_0(x) = a_0 - \frac{1}{4}a_1^2 - \frac{1}{2}a_1' \tag{6}$$

is a *semi-invariant* of (1), $\delta = b_1^2 - 4b_0$ is a *discriminant* of a characteristic equation

$$r^2 + b_1 r + b_0 = 0. \tag{7}$$

Then, the multiplier $v(x)$ of KL-transformation satisfies to the equation resulted

$$v(x) = |u|^{-1/2}\exp(-\frac{1}{2}\int a_1 dx)\exp(\frac{1}{2}b_1\int u dx) \tag{8}$$

and, moreover, $v(x)$ and $u(x)$ are related by a differential equation

$$v'' + a_1(x)v' + a_0(x)v - b_0 u^2 v = 0. \tag{9}$$

Lemma 1 allows to find constructive advices on finding $u(x)$.

Let us consider the equation

$$y_1'' + a_1 y_1' + a_{01} y_1 = 0, \quad a_{01} = a_0(x) - b_0 u^2(x) \tag{10}$$

resulted from (9) after the change $v \to y_1$. This equation and (1) have the same kernel $u(x)$ of the KL-transformation. Since the coefficient $a_0(x)$ due to the formula

$$a_0(x) = a_{01}(x) + b_0 u^2(x) \tag{11}$$

contains $u^2(x)$ as an additive part (up to the multiplicative constant $b_0 \neq 0$), then it is reasonable to choose candidates for $u(x)$ among the expressions $a_0(x)$ or $A_0(x)$ or their additive parts.

Then the function $w(x) = u^2(x)$ must satisfy to the equation

$$\frac{1}{4}\frac{w''}{w} - \frac{5}{16}(\frac{w'}{w})^2 - \frac{1}{4}\delta w = A_0. \tag{12}$$

Remark: in order to apply this statements, at least one of expressions δ or b_0 should be not zero.

2.2. Factorization

Let us say that the equation (1) *admits factorization* if its differential operator

$$L = D^2 + a_1 D + a_0, \ D = d/dx \tag{13}$$

may be represented as a product of first order operators

$$L = (D - \alpha_2)(D - \alpha_1), \quad \alpha_1 = \alpha_1(x), \alpha_2 = \alpha_2(x). \tag{14}$$

Here the differential analog of Viet formulas

$$a_1 = -(\alpha_1 + \alpha_2), \ a_0 = \alpha_1\alpha_2 - a_1' \tag{15}$$

remains solid.

Due to the G.Mammana's theorem, (1) always admits factorization.

Lemma 2. The equation (1) may be reduced to (4) by the transformation (2) and admits factorization

$$L = (D - \frac{v'}{v} - \frac{u'}{u} - r_2 u)(D - \frac{v'}{v} - r_1 u)y = 0 \tag{16}$$

where r_1, r_2 are roots of the characteristic equation (7).

Lemma 3. Zeroes (roots) of factorization may be represented in the form

$$\alpha_1 = -\frac{1}{2}\frac{u'}{u} - \frac{1}{2}a_1 + \frac{\sqrt{\delta}}{2}u, \quad \alpha_2 = \frac{1}{2}\frac{u'}{u} - \frac{1}{2}a_1 - \frac{\sqrt{\delta}}{2}u. \tag{17}$$

This formula connects "zeroes" of factorization with Kummer-Liouville transformation.

Let us now find the feedback of KL-transformation with "zeroes" of factorization. Using the arbitrariness of determining the characteristic roots r_1, r_2 , we may require $r_1 = r_2 = 0$. Then above formula takes a form

$$\alpha_1 = \frac{v'}{v}, \quad \alpha_2 = \frac{v'}{v} + \frac{u'}{u} \tag{18}$$

from where

$$v = e^{\int a_1 dx}, \quad u = e^{\int (\alpha_2 - \alpha_1) dx}. \tag{19}$$

So, we may formulate

Lemma 4. The equation (1) by transformation

$$y = e^{\int \alpha_1 dx}, \quad dt = e^{\int (\alpha_2 - \alpha_1) dx} dx \tag{20}$$

may be reduced to the equation with constant coefficients

$$z''(t) = 0. \tag{21}$$

Since in this case the KL-transformation is represented using the "roots" of factorization, the elementary procedure for finding factorization may be pointed.

Let us consider the Kummer-Schwarz equation (5) that due to (15) will take a form

$$A_0 = -\frac{1}{4}(\alpha_2 - \alpha_1)^2 - \frac{1}{2}(\alpha_2 - \alpha_1)' \tag{22}$$

from where

$$\alpha = \alpha_2 - \alpha_1.$$

So, we may find α as $\alpha = -2\sqrt{w}$, where w is an a additive part of semi-invariant A_0.

2.3. Fundamental system of solutions of LODE

Case 1: equation (1) admits the factorization (19).

Lemma 5. Equation (1) has the following fundamental system of solutions (FSS):

$$y_{1,2} = u^{-1/2} \exp(-\frac{1}{2}\int a_1 dx) \exp(\pm\frac{\sqrt{\delta}}{2}\int u dx), \quad \delta \neq 0, \tag{23}$$

$$y_1 = u^{-1/2} \exp(-\frac{1}{2}\int a_1 dx), \; y_2 = y_1 \int u dx, \quad \delta = 0. \tag{24}$$

Case 2: $b_1 = b_0 = 0$.

Lemma 6. If the factorization (14) is known, then FSS as follows:

$$y_1 = e^{\int \alpha_1 dx}, \quad y_2 = y_1 \int e^{\int (\alpha_2 - \alpha_1) dx} dx. \tag{25}$$

2.4. Partial solution

Let the nonhomogeneous equation

$$y'' + a_1 y' + a + 0y = f(x) \tag{26}$$

be given. If the FSS $\{y_1, y_2\}$ of correspondent equation (1) is known, then the partial solution y^* for Lemma 6 has the form

$$y^* = -y_1 \int e^{\int a_1 dx} y_2 f dx + y_2 \int e^{\int a_1 dx} y_1 f dx \tag{27}$$

while for Lemma 5

$$y^* = \frac{1}{2\sqrt{-b_0}}\Big(y_1 \int e^{\int a_1 dx} y_2 f dx - y_2 \int e^{\int a_1 dx} y_1 f dx\Big), \quad b_0 \neq 0. \tag{28}$$

2.5. Semi-invariants and special cases of a Kummer-Liouville transformation

2.5.1. Semi-invariant J_0 (invariant by the transformation of dependent variable).

As it was already mentioned above, J_0 has a form

$$J_0 = a_0 - \frac{1}{4}a_1^2 - \frac{1}{2}a_1'.$$

If $J_0 = const$, then the KL-transformation will take a form

$$y = \exp(-\frac{1}{2}\int a_1 dx) z, \quad dt = dx. \tag{29}$$

Factorization of operator L in this case will become commutative:

$$L = (D + \frac{1}{2}a_1 + \sqrt{b_0})(D + \frac{1}{2}a_1 - \sqrt{b_0}). \tag{30}$$

2.5.2. Semi-invariant J_1 (invariant by the transformation of independent variable).

$$J_1 = a_0 e^{2\int a_1 dx}(b_1 \int e^{-\int a_1 dx} dx + c)^2. \tag{31}$$

If $J_1 = const$, then the KL-transformation will take a form

$$y = z, \quad dt = -\frac{e^{-\int a_1 dx}}{b_1 \int e^{-\int a_1 dx} dx + c} dx. \tag{32}$$

We may determine whether J_1 is a constant or not, using the formula

$$\frac{a_1}{\sqrt{a_0}} + \frac{1}{2}\frac{a_0'}{a_0\sqrt{a_0}} = b_1 = const. \tag{33}$$

2.6. Equations solvable algebraically

2.6.1. Exponential solutions

Let the equation $Ly = 0$ have an exponential solution $y = e^{\lambda x}$, where $\lambda = const$. Then the characteristic equation

$$r^2 + a_1(x)r + a_0(x) = 0 \tag{34}$$

has among its roots r_1, r_2 not a function but a number λ:

$$r_{1,2} = -\frac{a_1}{2} \pm \sqrt{\frac{a_1^2}{4} - a_0}. \tag{35}$$

Factorization L takes a form:

$$L = (D + a_1 + \lambda)(D - \lambda). \tag{36}$$

2.6.2. Adjoint equations

By definition, the *adjoint* for $Ly = 0$ equation $L^*y = 0$ is an equation

$$L^*y \equiv y'' - a_1 y' + (a_0 - a_1')y = 0. \tag{37}$$

It admits factorization

$$L^* = (D + \alpha_1)(D + \alpha_2). \tag{38}$$

Let us consider the characteristic equation for adjoint one:

$$r^2 - a_1 r + a_0 - a_1' = 0. \tag{39}$$

If one of its roots is a number λ, then the factorization L^* takes a form:

$$L^* = (D - a_1 + \lambda)(D - \lambda). \tag{40}$$

Simultaneously, the factorization L is of the form

$$L = (D + \lambda)(D + a_1 - \lambda). \tag{41}$$

2.6.3. Exact equation

If $\lambda = 0$, then we have *an exact equation* , for which the factorization L has a form

$$L = D(D + a_1). \tag{42}$$

3. Implementation

The REDUCE program SOLDE which implements the algorithm described above, has the following main characteristics:

```
INPUT: a2,a1,a0,f              %coefficients of given equation
OUTPUT: u,v,                   %variable change
        b1,b0,                 %coefficients of reduced equation
        alfa1,alfa2,           %coefficients of factorization
        y1,y2,                 %FSS
        yp                     %partial solution for nonhomogeneous ODE
```

The method is, really, an algorithm itself, so the computer program SOLDE written in computer algebra system REDUCE, follows to the second part of this paper in detail, so there is no need to describe it here. It should be pointed only, that for make the program more convenient for users and faster the second author decided to include the coefficient a_2 under consideration. It may be explained by the fact that sometimes, that was proved theoretically, the candidate for $u(x)$ may be taken from $u(x) = 1/q(x)$, where $q(x)$ is a $a_2(x)$ or its multiplicative part.

Program SOLDE was tested on hundreds of equations and it had success in 75% cases. Failure in the left 25% are caused by two reasons: 1) the other important transformation, of Euler-Imshenetsky-Darboux, was not included to the program yet, though there already exist all the necessary formulas for its application here, and 2) the algorithm doesn't work when $b_1 = b_0 = 0$.

For illustrate the current possibilities of this program, let us name the equations from the demonstration file:

```
*** I.Equations with constant coefficients ***
       Y''+ A*Y'+ B*Y = F(X)
       Y''+ A*Y'+ A*A/4*Y = F(X)
*** II.Euler's equations ***
      Y'' + A/X * Y' + B/(X*X) * Y = F(X)/(X*X)
*** III.Equations of a form :    ***
       Y'' + A1(x) * Y' = F(X)
*** IY.Equations with exponential coefficients ***
       Y''+ A*Y'+ B*E**(2*A*X) * Y = F(X)
*** Y.Equations with trigonometrical functions ***
       Y''+ 2*A*COT(A*X) * Y'+ (B*B-A*A) * Y = 0
       Y''+ (M*M + A/SIN(2*M*X)**2) * Y = 0
*** YI.Equations with hyperbolic functions ***
```

```
        Y''+ 2*TANH(X) * Y' + B*Y = 0
        Y''+ (-M*M + A/(SINH(M*X)**4)) * Y = 0
*** YII.Equations with algebraic coefficients ***
        Y'' + 8*A/(5*(A*X+B))*Y'+C*(A*X+B)**(1/5)/(5*(A*X+B))*Y = 0
*** YIII.Equations with mixed coefficients ***
        Y'' + 2*A/X*Y' +((B*B*E**(2*C*X)-1/4)*C*C+A*(A-1)/(X*X)) * Y = 0
        Y'' + (1/(4*X*X) + 1/(X*X)/(P*LOG(X)+Q)**4)) * Y = 0
*** IX.Equations with arbitrary functions ***
       Y'' + 2*F(X)*Y' + (F(X)*F(X)+F'(X)+G''(X)/2/G(X)-3/4*G'(X)*G'(X)/
            G(X)*G(X)-A*G(X)*G(X))*Y=0
*** X.Equations with rational coefficients ***
      Y'' + D/(A*X*X+B*X+C)**2 * Y = 0
      Y''+(-M*(M+1)/(X*X) +(1/(-P/(2*M+1)*X**(-M)+Q*X**(M+1))**4)*Y=0
```

High efficiency of this implementation is caused by the fact that entire package was developed around the procedure SOLDE which allows to the user to investigate his ODE from different sides, and even the program doesn't give an answer itself, it often helps the user in finding the approaches to his equation making more effective and faster his operations and testing his propositions. Thus, each ODE may be investigated during the whole session by the procedures shown in Table 1.

Table 1. Procedures in the package.

SOLDE(a2, a1, a0, f)	solve the equation and display SUMMARY; automatically put the new equation
VERFAC(alfa1) VERSOL(y1) VERTRANS(u, v)	verify the given factorization, and if it is correct, display SUMMARY; similar to VERFAC - for solution similar to VERFAC - for transformation
PUTEQ(a2, a1, a0, f, J0)	put the new equation without its solving by the procedure SOLDE; normally you would enter J0=0 and then semi-invariant would be computed by program; if you anywhy aren't satisfied from its value, you enter value of it yourself.
NORM(W)	normalize W, i.e., to delete constant factor
SINV()	display the meanings of semi-invariants
SUMMARY()	display all available information on current equation,including
HELP() INFO()	display brief and complete information on the package

Example of this program in action is given below:

```
2: %Special case of this equation with a=1 was solved by Kovacic [5]
   %in another way
solde(x^2,0,-a*x+3/16,0);
*Summary of the operations*
**************************************************************
                             2                    16*A*X - 3
*The equation was:      (X )*y''+ (0)*y'+ ( - ------------)*y = 0
                                                      16
                                                      16*A*X - 3
*The semi-invariant by dependent variable: J0=  - ------------
                                                          2
                                                      16*X

                                 1/4                  1
*The transformation:     y = (X     )*z,   dt = (---------) dx
                                                 SQRT(X)
*  leads to    z''(t)+(0)*z'(t)+ ( - A)*z(t) = 0
*The factorization:
                         4*SQRT(A)*X + SQRT(X)
*   L = (D - ( - -----------------------)) (D - (
                            4*SQRT(X)*X
 4*SQRT(A)*X + SQRT(X)
-----------------------))
      4*SQRT(X)*X
*Fundamental system of solutions of Ly=0:
         1/4  2*SQRT(X)*SQRT(A)
   Y1= X    *E
                  1/4
                 X
  Y2= --------------------
        2*SQRT(X)*SQRT(A)
       E
**************************************************************
```

The previous version of the program is described in [6] .

References

[1] Berkovich L.M., *Factorization and Transformations of Ordinary Differential Equations*, Saratov Univ. Publishers, Saratov, 1989 (in Russian)

[2] Singer M. F., *Formal Solutions of Differential Equations*, J. Symb. Comp., 1990, 10, 59-94

[3] Borůvka O., *Lineare Differential Transformationen 2. Ordnung*, Berlin, 1967

[4] Berkovich L. M., *Transformations of Linear Ordinary Differential Equations*, Kuibyshev State Univ, 1978 (in Russian)

[5] Kovacic J., *An Algorithm for Solving Second Order Linear Homogeneous Differential equations*, J. Symb. Comp., 1986, 2, 1

[6] Berkovich L. M., Gerdt V. P., Kostova Z. T., Nechaevsky M. L., *Second Order Reducible Linear Differential Equations*, Preprint JINR E5-89-141, Dubna, 1989.